REMARKS

35 U.S.C. § 103 Claim Rejection

By the Office Action dated April 15, 2009, the Examiner has rejected claim 16 under 35 U.S.C. § 103(a) as being unpatentable over James (U.S. Patent Application Publication Number 2003/0217169) (hereinafter "James") in view of Challenger (U.S. Patent No. 6,026,413) (hereinafter "Challenger"). In order to form a proper obviousness rejection of a claim under 35 U.S.C. § 103(a), a collection of references together must teach or suggest each element of the claim, including the relationships between the elements. If any element is not fully taught by the combined references, the rejection cannot be sustained.

The Examiner asserted that,

> James discloses substantially all of the elements, in figures 1-3, in a client-server computing system having a cache and storing eXtensible Markup Language (XML) data as data objects, a method for determining invalid cached objects comprising
> transforming XML data into a format suitable for a client application based on a set of transformation rules ("XML formatted documents which can be transformed into other formats according to instructions included in corresponding XSL stylesheet documents", paragraph [0019]);
> determining dependencies between cached objects and XML data related to the cached objects ("Based upon the freshness requirements of the XML document and the XSL document, it can be determined whether located transformed content satisfies the freshness requirements. If so, the located transformed content can be returned to the client process 102 over the public network 104", paragraph [0022])".

(See Office Action, pages 2-3.) The Examiner admitted that

James discloses substantially all of the elements, in figures 1-3, in a client-server computing system having a cache and storing eXtensible Markup Language (XML) data as data objects, a method for determining invalid cached objects . . .

except monitoring updates to the related XML data; and

determining the cached objects that are affected by changes to the related data based on the dependencies,

where data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure.

(See Office Action, pages 2-3.)

The Examiner then asserted that, for claim 16,

Challenger teaches in figures 1, 5, 12, 18, and 27-30, the cache manager 1 keeps track of these dependencies[,] . . . [w]henever a process modifies state which could affect the value of a complex object in a cache, the process specifies the underlying data which it is updating, col. 8, lines 54-67 and col. 9, lines 1-6, col. 8, (monitoring updates to the related data; and determining the cached objects that are affected by changes to the related data based on the dependencies), a directed graph in which each node is a leaf node r1 . . . r3 or a maximal node co1, co2[,] . . .[r]ecall that a leaf node is a node which is not the target of any edges and a maximal node is a node which is not the source of any edges, col.9, lines 7-25 (where data is represented as a tree structure having a plurality of nodes) and The application also uses APIs to notify the object manager 1 of records r1 . . . r4 which have changed[,] . . .[and] [w]hen the cache manager 1 is notified of changes to a record r1 . . . r4, it must identify which complex objects co1, co2 have been affected and cause the identified complex objects to be deleted (or updated) from any caches 2 containing them, col. 9, lines 26-38, (and the cached objects that are

affected by the data changes are determined using the tree structure).

(See Office Action, page 3.) The Examiner finally asserted that, for claim 16, "[i]t would have been obvious to one having ordinary skill in the art at the time the invention was made to reduce the overhead dynamic data of an object dependence graph (node, tree) set of updates to caches consistently by managing dependencies." (See Office Action, page 3.)

To the extent the Examiner's language at pages 2 and 3 of the Office Action can be understood, it appears that the Examiner has asserted the following correspondence between James and Challenger and claim 16:

Claim 16	**James**	**Challenger**
16. In a client-sever computing system having a cache and storing eXtensible Markup Language (XML) data as data objects, a method for determining invalid cached objects comprising:	James does not teach this claim element.	Challenger does not teach this claim element.
transforming XML data into a format suitable for a client application based on a set of transformation rules;	James does not teach this claim element.	Challenger does not teach this claim element.
determining dependencies between cached objects and XML data related to the cached objects;	James does not teach this claim element.	Challenger does not teach this claim element.
monitoring updates to the related XML data;	James does not teach this claim element.	Challenger does not teach this claim element.

and determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and* *the cached objects that are affected by the data changes are determined using the tree structure.*	James does not teach this claim element.	Challenger does not teach this claim element.

In reviewing the cited portions of James and Challenger, however, it becomes apparent that James and Challenger have been generalized, and, in fact, does not support the position asserted by the Examiner.

monitoring updates to the related *XML* data

Also, James and Challenger, alone or in combination, fail to teach or suggest "monitoring updates to the related *XML* data", as required by claim 16. The Examiner admitted that "James discloses substantially all of the elements, in figures 1-3, in a client-server computing system having a cache and storing eXtensible Markup Language (XML) data as data objects, a method for determining invalid cached objects . . . except monitoring updates to the related XML data". (See Office Action, pages 2-3.) Thus, the Examiner admitted that James fails to teach or suggest "monitoring updates to the related *XML* data". Challenger discloses that "a cache manager 1 (which is an example of an object manager) determines how changes to underlying data affect the values of objects."

(See Challenger, column 8, lines 54-56.) Thus, Challenger fails to teach or suggest that cache manager 1 can be used for "monitoring updates to the related *XML* data". Therefore, Challenger cannot teach or suggest "monitoring updates to the related *XML* data". Therefore, James and Challenger, alone or in combination, cannot teach or suggest the claim 16 element of "monitoring updates to the related *XML* data".

determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure*

In addition, James and Challenger, alone or in combination, fail to teach or suggest "determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure*", as required by claim 16. The Examiner admitted that "James discloses substantially all of the elements, in figures 1-3, in a client-server computing system having a cache and storing eXtensible Markup Language (XML) data as data objects, a method for determining invalid cached objects . . . except . . . determining the cached objects that are affected by changes to the related data based on the dependencies, where data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure. (See Office Action, pages 2-3.) Thus, the Examiner admitted that James fails to teach or suggest "determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure*". Challenger discloses an "object manager [that] maintains the data dependence information . . . [such that] [w]henever new objects are created or the data dependencies change, the object manager is responsible for updating the appropriate information." (See Challenger, column 4, lines 33-37.) Challenger also discloses that "a cache manager 1 (which is an example of an object manager) determines how changes to underlying data affect the values of objects." (See Challenger, column 8, lines 54-56.) Thus, Challenger fails to teach or suggest that either the object manager or

cache manager 1 can be used for "determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure*". Therefore, Challenger cannot teach or suggest "determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure*". Therefore, James and Challenger, alone or in combination, cannot teach or suggest the claim 16 element of ""determining the cached objects that are affected by changes to the related XML data based on the dependencies, *wherein data is represented as a tree structure having a plurality of nodes and the cached objects that are affected by the data changes are determined using the tree structure*". It is therefore clear that James and Challenger, alone or in combination, cannot teach or suggest each element of claim 16 and, therefore, a rejection of claim 16 under 35 U.S.C. § 103(a) would be inappropriate.

Conclusion

It is therefore clear that claim 16 complies with the requirements of 35 U.S.C. §§ 101, 102, 103, and 112. The application is therefore in condition for allowance. Early notification to that effect is respectfully solicited.

In the event that any issue remains unresolved, the Examiner is invited to telephone the undersigned at 408-927-3377.

Respectfully Submitted,
/Leonard T. Guzman/

Date: June 15, 2009

Leonard T. Guzman
Reg. No. 46,308

IBM Almaden Research Center
650 Harry Road
C45A/J2B
San Jose, CA 95120
Phone Number: 408-927-3377 Facsimile Number: 408-927-3375